

File 8: Ei Compendex(R) 1970-2003/Nov W2  
(c) 2003 Elsevier Eng. Info. Inc.  
File 35: Dissertation Abs Online 1861-2003/Oct  
(c) 2003 ProQuest Info&Learning  
File 202: Info. Sci. & Tech. Abs. 1966-2003/Nov 17  
(c) 2003 EBSCO Publishing  
File 65: Inside Conferences 1993-2003/Nov W3  
(c) 2003 BLDSC all rts. reserv.  
File 2: INSPEC 1969-2003/Nov W2  
(c) 2003 Institution of Electrical Engineers  
File 233: Internet & Personal Comp. Abs. 1981-2003/Jul  
(c) 2003, EBSCO Pub.  
File 94: JICST-EPlus 1985-2003/Nov W3  
(c) 2003 Japan Science and Tech Corp(JST)  
File 603: Newspaper Abstracts 1984-1988  
(c) 2001 ProQuest Info&Learning  
File 483: Newspaper Abs Daily 1986-2003/Nov 17  
(c) 2003 ProQuest Info&Learning  
File 6: NTIS 1964-2003/Nov W3  
(c) 2003 NTIS, Intl Cpyrght All Rights Res  
File 144: Pascal 1973-2003/Nov W2  
(c) 2003 INIST/CNRS  
File 434: SciSearch(R) Cited Ref Sci 1974-1989/Dec  
(c) 1998 Inst for Sci Info  
File 34: SciSearch(R) Cited Ref Sci 1990-2003/Nov W2  
(c) 2003 Inst for Sci Info  
File 99: Wilson Appl. Sci & Tech Abs 1983-2003/Oct  
(c) 2003 The HW Wilson Co.  
File 583: Gale Group Globalbase(TM) 1986-2002/Dec 13  
(c) 2002 The Gale Group  
File 266: FEDRIP 2003/Sep  
Comp & dist by NTIS, Intl Copyright All Rights Res  
File 95: TEME-Technology & Management 1989-2003/Nov W1  
(c) 2003 FIZ TECHNIK  
File 438: Library Lit. & Info. Science 1984-2003/Oct  
(c) 2003 The HW Wilson Co

Set	Items	Description
S1	18542	(SEARCH??? OR QUER??? OR QUERY??? OR RETRIEV?)(10N)(SIMULT-ANEUOUS? OR CONCURREN? OR COINCIDENT? OR PARALLEL)
S2	189168	(ADD??? OR INCREAS??? OR SUPPLEMENT? OR ENLARG? OR EXPAND? - ?? OR INCREMENT?)(5N)(PARTITION? ? OR GROUP???? OR SET? ? OR - CLUSTER? ? OR COLLECTION? ? OR DATABASE? ? OR DATA()BASE? ? OR REPOSITOR??? OR SEARCH???)
S3	31093	(ADD??? OR INCREAS??? OR SUPPLEMENT? OR ENLARG? OR EXPAND? - ?? OR INCREMENT?)(5N)(QUERY??? OR QUERIE? ? OR CONTAINER? ? OR BUCKET? ? OR INSTANCE? ? OR COPIE? ? OR COPY OR REPLICAT? OR DUPLICAT? OR REPRODUC? OR FACSIMILE? ?)
S4	722256	TREE? ? OR HIERARCH?
S5	16219	SEARCH()ENGINE? ?
S6	566	S1 AND S2:S3
S7	180617	(ADD??? OR INCREAS??? OR SUPPLEMENT? OR ENLARG? OR EXPAND? - ?? OR INCREMENT?)(5N)(PARTITION? ? OR GROUP???? OR SET? ? OR - CLUSTER? ? OR COLLECTION? ? OR DATABASE? ? OR DATA()BASE? ? OR REPOSITOR???)
S8	29067	(ADD??? OR INCREAS??? OR SUPPLEMENT? OR ENLARG? OR EXPAND? - ?? OR INCREMENT?)(5N)(CONTAINER? ? OR BUCKET? ? OR INSTANCE? ? OR COPIE? ? OR COPY OR REPLICAT? OR DUPLICAT? OR REPRODUC? OR FACSIMILE? ?)
S9	169	S1 AND S7:S8
S10	129	RD (unique items)
S11	103	S10 NOT PY=2000:2003
S12	16	S11 AND S4:S5
S13	87	S11 NOT S12
S14	36	S13 AND SEARCH?
S15	51	S13 NOT S14

12/5/2 (Item 2 from file: 8)

DIALOG(R)File 8:EI Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

00961895 E.I. Monthly No: EI8011081297 E.I. Yearly No: EI80016248

Title: PARALLEL TREE SEARCH METHOD.

Author: Nakagawa, Sei-ichi; Sakai, Toshiyuki

Corporate Source: Kyoto Univ, Jpn

Source: IJCAI-79, Proc of the Int Jt Conf on Artif Intell, 6th, v 2, Tokyo, Jpn, Aug 20-23 1979 Publ by Int Jt Conf on Artif Intell, 1979. Available from Stanford Univ, Comput Sci Dep, Calif p 628-632

Publication Year: 1979

Language: ENGLISH

Journal Announcement: 8011

Abstract: A new **tree** search method is described which **searches** best few paths with backtracking in **parallel** by the following algorithm ( ALPHA - BETA - GAMMA **search** ). Expand the best ALPHA nodes at one time. If the number of new nodes which have been expanded by a selected node exceeds a pre-set threshold ( BETA ), only the best BETA nodes are kept and others are pruned. If the total number of newly generated node sequences and non-**expanded** node sequences exceeds a pre-**set** threshold ( GAMMA ), only the best GAMMA node sequences are kept, and others are removed. Repeat these consecutive processes until a complete sequence is generated. This method was evaluated by simulations and experiments, and good performance was obtained both in the search efficiency and in the recognition rate. 12 refs.

Descriptors: \*COMPUTER PROGRAMMING--\*Subroutines

Classification Codes:

723 (Computer Software)

72 (COMPUTERS & DATA PROCESSING)

12/5/3 (Item 3 from file: 8)

DIALOG(R)File 8:EI Compendex(R)

(c) 2003 Elsevier Eng. Info. Inc. All rts. reserv.

00348775 E.I. Monthly No: EI7402007440

Title: MASTER LINKS -- A HIERARCHICAL DATA SYSTEM.

Author: Gibson, T. A.; Stockhausen, P. F.

Source: Bell System Technical Journal v 52 n 10 Dec 1973 p 1691-1724

Publication Year: 1973

CODEN: BSTJAN ISSN: 0005-8580

Language: ENGLISH

Journal Announcement: 7402

Abstract: Master Links is a software system used to build, administer, and access **hierarchical** data bases. It is designed to operate in a time-sharing environment, and, in particular, it allows multiple **concurrent** updates and **retrievals** on the same data base. A Build module is used to specify the **hierarchical** configuration of a data base and an initial " storage mapping " of the elements of the **hierarchy** into a particular file layout. A set of administrative routines is provided for altering the mapping and other such maintenance purposes. The access routines have three levels of interface, from primitive and flexible to sophisticated and functional. The interfaces are all defined in terms of the **hierarchical** structure and independent of the storage mapping. Thus, an alteration of the storage mapping for a data base does not require changing any programs that access data using these interfaces. The lowest-level interface enables the calling program to **add** to the **data base** , update a value, or retrieve a value, in terms of a **hierarchy** position. The second-level interface facilitates traversal of a **hierarchy** by enabling the calling program to specify portions of the **hierarchy** over which a process is to operate. Such a specification, called an " access **tree** , " consists of data which can be generated at execution time by the calling routine. As in the first level, data are transferred one at a time. The third-level interface is a function evaluation mechanism which computes values from data base values and other computed values according to function definitions passed to it at execution time. Like an access **tree** , a function definition is itself data which can be constructed at execution

time by the client process.

Descriptors: \*INFORMATION RETRIEVAL SYSTEMS

Classification Codes:

723 (Computer Software); 901 (Engineering Profession)

72 (COMPUTERS & DATA PROCESSING); 90 (GENERAL ENGINEERING)

12/5/5 (Item 1 from file: 202)

DIALOG(R)File 202:Info. Sci. & Tech. Abs.

(c) 2003 EBSCO Publishing. All rts. reserv.

3304076

**Northern Light bridges two worlds: innovative search service expands possibilities for ordinary Web users.**

Author(s): O'Leary, Mick

Information Today vol. 15, no. 4, pages 15-18

Publication Date: April 1998

ISSN: 8755-6286

Language: English

Document Type: Journal Article

Record Type: Abstract

Journal Announcement: 3308

Presents a favorable review of Northern Light (NL), a **search** service from Northern Light of Cambridge, MA. Notes the product **simultaneously searches** the Web and a large collection of proprietary databases, using enhanced search and indexing features. Suggests NL is comparable to AltaVista or Hotbot as a comprehensive Web **search engine**, but **adds** the Special **Collection** proprietary **database** feature sets this product apart. Notes the Special Collection has full-text content from almost 3,000 publications, licensed from leading database producers. Says the product has a daring business model, with charges based entirely on output. Adds the timeliness could be improved for current event topics. Concludes, "NL is a visionary yet well-designed product whose future is built on faith."

Descriptors: Online retrieval; Online retrieval

Classification Codes and Description: 5.11 (Searching and Retrieval)

Main Heading: Information Processing and Control

12/5/7 (Item 1 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

6662987 INSPEC Abstract Number: C2000-09-7250N-014

**Title: Integration of multiple heterogeneous databases using a client-side meta search agent**

Author(s): Taehee Kim; Seonho Kim; Sun-Hwa Hahn; Jin-Hyung Kim

Author Affiliation: Korea R&D Inf. Center, Taejeon, South Korea

Conference Title: Proceedings of the 1st Asia-Pacific Conference on IAT. Intelligent Agent Technology Systems, Methodologies, and Tools p.234-43

Editor(s): Liu, J.; Zhong, N.

Publisher: World Scientific, Singapore

Publication Date: 1999 Country of Publication: Singapore xiii+505 pp.

ISBN: 981 02 4054 6 Material Identity Number: XX-1999-02419

Conference Title: Proceedings of the 1st Asia-Pacific Conference on Intelligent Agent Technology

Conference Date: 14-17 Dec. 1999 Conference Location: Hong Kong

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P); Experimental (X)

Abstract: Low cost information access and the high coverage of information sources would increase research productivity. Integrating multiple information sources, or **databases** would **increase** both information accessibility and information completeness. This paper introduces an approach to building a meta **search** agent that **simultaneously retrieves** information of users' request from multiple heterogeneous databases on the World Wide Web. Multiple heterogeneous databases appear to be integrated from the users' perspectives. While most

conventional meta search agents are of server-side type, our meta search agent is of client-side type that runs on client computers for distributed use of resources. Our experiment showed that the client-side meta search agent is slightly better in response time in comparison with the conventional meta search agent. In particular, it is significantly less sensitive to the busy network time compared to server-side meta search agents. (17 Refs)

Subfile: C

Descriptors: client-server systems; distributed databases; information resources; information retrieval; Internet; **search engines**; software agents

Identifiers: multiple heterogeneous database integration; client-side meta search agent; low cost information access; information sources; information accessibility; information completeness; information retrieval; World Wide Web; server-side type; client-side type; experiment; response time

Class Codes: C7250N (Search engines); C6160B (Distributed databases); C6170 (Expert systems and other AI software and techniques); C7250R (Information retrieval techniques); C7210N (Information networks)

Copyright 2000, IEE

12/5/8 (Item 2 from file: 2)

DIALOG(R)File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

6604285 INSPEC Abstract Number: C2000-07-6160J-006

**Title: High-performance extensible indexing**

Author(s): Kornacker, M.

Author Affiliation: California Univ., Berkeley, CA, USA

Conference Title: Very Large Data Bases. Proceedings of the Twenty-Fifth International Conference on Very Large Data Bases p.699-708

Editor(s): Atkinson, M.; Orlowska, M.E.; Valduriez, P.; Zdonik, S.; Brodie, M.

Publisher: Morgan Kaufmann Publishers, Orlando, FL, USA

Publication Date: 1999 Country of Publication: USA xviii+761 pp.

Material Identity Number: XX-1999-02812

Conference Title: Proceedings of 25th International Conference on Very Large Databases

Conference Sponsor: Oracle; Sun Microsystems.; IBM; Microsoft SQLServer7.0; Scottish Widows

Conference Date: 7-10 Sept. 1999 Conference Location: Edinburgh, UK

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: Today's object-relational DBMS (ORDBMS) are designed to support novel application domains by providing an extensible architecture, **supplemented** by domain-specific **database** extensions supplied by external vendors. An important aspect of ORDBMS is support for extensible indexing, which allows the core database server to be extended with external access methods (AM). This paper describes a new approach to extensible indexing implemented in the Informix Dynamic Server with Universal Data Option (IDS/UDO). The approach is based on the generalized search **tree**, or GiST, which is a template index structure for abstract data types that supports an extensible set of **queries**. GiST encapsulates core database indexing functionality including **search**, update, **concurrency** control and recovery, and thereby relieves the external access method (AM) of the burden of dealing with these issues. The IDS/UDO implementation employs a newly designed GiST API that reduces the number of user defined function calls, which are typically expensive to execute, and at the same time makes GiST a more flexible data structure. Experiments show that GiST-based AM extensibility can offer substantially better performance than built-in AM when indexing user-defined data types.

(12 Refs)

Subfile: C

Descriptors: abstract data types; application program interfaces; concurrency control; database indexing; object-oriented databases; query formulation; query processing; relational databases; software performance evaluation; **tree** data structures; **tree** searching; very large databases

Identifiers: high-performance extensible indexing; object-relational DBMS ; ORDBMS; domain-specific database extensions; core database server; external access methods; Informix Dynamic Server with Universal Data Option ; IDS/UDO; generalized search **tree** ; GiST; template index structure; abstract data types; queries; database indexing; database searching; database updating; concurrency control; database recovery; very large databases; API; user defined function calls; performance; user-defined data types

Class Codes: C6160J (Object-oriented databases); C6160D (Relational databases); C6160Z (Other DBMS); C6120 (File organisation)

Copyright 2000, IEE

12/5/10 (Item 4 from file: 2)

DIALOG(R) File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

04304013 INSPEC Abstract Number: C9301-6160B-031

**Title: Design and implementation of parallel database processing on a shared memory multiprocessor system**

Author(s): Satoh, T.; Hirano, Y.; Honishi, T.; Inoue, U.

Author Affiliation: NTT Network Inf. Syst. Labs., Tokyo, Japan

Conference Title: Future Databases '92. Proceedings of the Second Far-East Workshop on Future Database Systems p.337-46

Editor(s): Qiming Chen; Yahiko Kambayashi; Sacks-Davis, R.

Publisher: World Scientific, Singapore

Publication Date: 1992 Country of Publication: Singapore xii+418 pp.

ISBN: 981 02 1040 X

Conference Date: 26-28 April 1992 Conference Location: Kyoto, Japan

Language: English Document Type: Conference Paper (PA)

Treatment: Practical (P)

Abstract: The time required to execute batch transactions is becoming longer because of the **increasing** size of **databases** and because batch transaction queries are becoming more complex. Batch transactions include range queries executed with and without the use of indexes, join queries and table merge operations which include index updating. Execution time can be reduced by introducing intra-parallelism in the processing of these transactions. The authors propose a load balancing algorithm and an index structure on a shared memory multiprocessor system to achieve highly scalable intra-parallelism in executing batch transactions. In the proposed algorithm, the number of allocated tasks is dynamically calculated on the basis of the number of remaining tasks and the max/min processing time of each task to decrease the overhead of parallel execution. The proposed index structure is a hybrid structure consisting of some B- **trees** associated with a hashing function. Highly **parallel** execution of range **queries** and table merges requiring index operations is achieved with this hybrid structure. The effects of the adaptive load balancing and hybrid index structure are evaluated by a prototype DBMS implemented in a shared memory multiprocessor. The performance evaluation shows the adaptive algorithm can increase the scalability by 0.1 to 0.2. Range queries and table merging requiring index operations can be easily executed in parallel by using the hybrid index structure. (16 Refs)

Subfile: C

Descriptors: distributed databases; indexing; multiprocessing programs; query processing; resource allocation; shared memory systems

Identifiers: parallel database processing; shared memory multiprocessor system; batch transactions; range queries; indexes; join queries; table merge operations; index updating; intra-parallelism; load balancing algorithm; allocated tasks; B- **trees** ; hashing function; performance evaluation; scalability; table merging

Class Codes: C6160B (Distributed DBMS); C6150N (Distributed systems)

12/5/11 (Item 5 from file: 2)

DIALOG(R) File 2:INSPEC

(c) 2003 Institution of Electrical Engineers. All rts. reserv.

04275113 INSPEC Abstract Number: C9212-5260B-093

**Title:** A database approach to hierarchical image storage and recognition  
**Author(s):** Al-Mouhamed, M.  
**Author Affiliation:** King Fahd Univ. of Pet. & Miner., Dhahran, Saudi Arabia  
**Conference Title:** Computer and Information Sciences VI. Proceedings of the 1991 International Symposium p.991-1000 vol.2  
**Editor(s):** Baray, M.; Ozguc, B.  
**Publisher:** Elsevier, Amsterdam, Netherlands  
**Publication Date:** 1991 **Country of Publication:** Netherlands **2 vol.** xviii+1249 pp.  
**ISBN:** 0 444 89067 X  
**Conference Date:** 30 Oct.-2 Nov. 1991 **Conference Location:** Side, Turkey  
**Language:** English **Document Type:** Conference Paper (PA)  
**Treatment:** Practical (P)  
**Abstract:** The paper presents the design and implementation of a **hierarchical** image storage and recognition system by using a database approach. The low level image modelling of planar objects, is converted into a set of strongly (corners) and slightly (segments) curved contours whose boundaries are found using constant curvature criterion. The partitioning enables generation of coarse and detailed models which include structural geometric information of the object. A set of descriptors is extracted from the coarse description in order to enable **parallel search** on indexed files to find out those classes that share the same descriptor. To recognize an object, the generated classes are sorted in the order of non- **increasing** frequency and a sub- **set** of these classes are used to evaluate the distance between the object and that class. Evaluation shows the ability to recognize entirely and partially observed object among a reasonably large dictionary. It is shown that the database approach to pattern recognition can be efficient for real-time application and greatly simplifies the design. (9 Refs)  
**Subfile:** C  
**Descriptors:** computerised pattern recognition; computerised picture processing; database management systems  
**Identifiers:** image recognition; **hierarchical** image storage; database approach; low level image modelling; planar objects; curved contours; partitioning; structural geometric information; **parallel search** ; indexed files; real-time application  
**Class Codes:** C5260B (Computer vision and picture processing); C1250 (Pattern recognition); C6160S (Spatial and pictorial databases)